DOCUMENT RESUME

ED 443 106 CS 014 065

AUTHOR Friel, Brian M.; Kennison, Shelia M.
TITLE Displaying Special Characters and Symbols in Computer-Controlled Reaction Time Experiments.
PUB DATE 1999-00-00
NOTE 24p.
PUB TYPE Reports - Research (143)
EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS *Computer Graphics; Computer Software; *Editing; Higher Education; *Reaction Time
IDENTIFIERS *Priming Effects; University of Oklahoma

ABSTRACT

A procedure for using MEL2 (Version 2.0 of Microcomputer Experimental Laboratory) and FontWINDOW to present special characters and symbols in computer-controlled reaction time experiments is described. The procedure permits more convenience and flexibility than in tachistocopic and projection techniques. FontWINDOW allows researchers to design fonts and edit special characters, which can be presented to a cathode ray tube (CRT) using MEL2 to control presentation and reaction time recording. FontWINDOW is a software package that allows the experimenter to very easily customize stimuli to fit the study's needs; MEL2 allows the experimenter the opportunity to present a wide variety of computer-controlled experiments, included, but not limited to, sentence processing, categorization, and priming experiments. A successful implementation involving two graphics fonts in a masked repetition priming experiment is described. Subjects, 38 undergraduate psychology students at the University of Oklahoma, were presented with letter strings for which they were instructed to make a lexical decision. The targets were primed with English words, which were either identical or unrelated to the target. (Contains a table and 22 references. An appendix shows procedure characteristics.) (NKA)

Reproductions supplied by EDRS are the best that can be made
from the original document.

ED 443 106

Displaying Special Characters and Symbols in Computer-Controlled Reaction Time Experiments

Brian M. Friel and Shelia M. Kennison

University of Oklahoma

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☐ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

CS 014 065

S. Kennison

BEST COPY AVAILABLE

Abstract

A procedure for using MEL2 and FontWINDOW to present special characters and symbols in computer-controlled reaction time experiments is described. The procedure permits more convenience and flexibility than in tachistoscopic and projection techniques. FontWINDOW allows researchers to design fonts and edit special characters, which can be presented to a cathode ray tube (CRT) using MEL2 to control presentation and reaction time recording. A successful implementation involving two graphics fonts in a masked repetition priming experiment is described.

Using FontWINDOW and MEL2 to Present Special Characters and Symbols in Computer-Controlled Reaction Time Experiments

Researchers interested in the word recognition process and the nature of lexical representation have used a variety of techniques utilizing reaction time measures (Rayner & Pollatsek, 1989). One such technique is the priming paradigm. In this paradigm, participants are presented with a target word and asked to perform a task, such as decide whether a letter string is a word (lexical decision task) or name a word or letter string (naming task). This task is typically preceded by a prime word that is either related or unrelated to the target. When the prime word is identical to the target, naming and lexical decision times tend to be facilitated. This type of priming as been referred to as repetition priming (Forbach, Stanners, & Hochhaus, 1974; Kirsner & Smith, 1974; Scarborough, Cortese, & Scarborough, 1977). There have been two main perspectives in explaining this facilitation, known as a repetition priming effect. One perspective maintains that repetition priming occurs because the prime accesses its lexical representation, leaving it in a temporary activated state when the target is presented. This residual activation allows recognition of the target to proceed more quickly. The other perspective maintains that the priming effect is due to the prime word forming an episodic trace, which is accessed during presentation of the target (see Forster & Davis, 1984, for a discussion on these perspectives).

In an effort to resolve this debate, researchers developed the masked priming paradigm to measure automatic processes more directly (cf. Evett & Humphreys, 1981; Forster & Davis, 1984). The masked priming paradigm is similar to classical repetition priming experiments in that a target word, written in uppercase letters, is preceded by a prime word, which is either identical or unrelated to the target. However, the prime is preceded by a forward masking stimulus (usually a row of hash marks, i.e., #########, or a random letter string), usually written

in lower case letters, and presented for an extremely short duration (usually for 60 ms or less). This manipulation of the prime's presentation typically renders it unavailable for conscious report. Thus, the chances of an episodic trace being laid down for the prime are minimized. Nevertheless, strong repetition priming effects in masked priming experiments are often found.

Researchers have typically utilized either tachistoscopic (or tachistoscopic-like) techniques or have utilized computer-controlled presentation for repetition priming studies. Computer controlled presentation offers obvious advantages over the tachistoscopic presentation. For one, computers are readily available in most laboratories, whereas tachistoscopes tend to be more difficult to come by. This is largely due to the fact that computers can be programmed to perform a much wider range of functions in the laboratory. Within a given experiment, computers can be programmed to store experimental stimuli, record data, carry out the appropriate statistical analyses, and control the presentation rate of stimuli. This latter capability is essential for conducting experiments using the masked priming paradigm because primes must be presented for extremely short durations. Computer-controlled presentation, when the computer is programmed correctly, also minimizes human error with respect to presentation order and data recording.

Another advantage provided by computers is that, given the appropriate software, manipulation the appearance of stimuli can be carried out more easily and with more precision than with traditional methods. One implication of this capability is that stimuli from other languages using scripts other than the Roman alphabet (e.g., Arabic, Chinese, Cyrillic, Hebrew, etc.) can be created and presented. Until recently, computer-controlled presentation of such stimuli could only be carried out with great difficulty, because most fonts that computers had access to were restricted to the Roman alphabet. Presenting non-Roman alphabetic scripts is an

important capability for researchers interested in the knowledge representations of these languages and for those who study bilinguals with knowledge of languages using two different scripts (e.g., Brown, Sharma, & Kirsner, 1984; Chen & Ng, 1989; Cheung & Chen, 1998; Gollan, Forster, & Frost, 1997; Keatley, Spinks, & DeGelder, 1994; Kirsner, Smith, Lockhart, King, & Jain, 1984).

The purpose of this paper is to describe a technique for using FontWINDOW, Version 4.1 (Metagraphics Software, Carmel, CA) to present customized fonts with Version 2.0 of Microcomputer Experimental Laboratory (MEL2; Schneider, 1995; Psychological Software Tools, Pittsburgh). MEL2 allows the experimenter the opportunity to present a wide variety of computer-controlled experiments, including, but not limited to, sentence processing, categorization, and priming experiments. FontWINDOW is a software package that allows the experimenter to very easily customize stimuli to fit the study's needs. Therefore, this paper should be of interest to researchers who wish to take advantage of the precision that computer-controlled experiments provide, while at the same time having the flexibility of creating their own stimuli. This technique should be particularly beneficial to researchers who wish to use two or more different fonts, special characters, or non-Roman scripts. A masked repetition priming experiment is described to demonstrate the technique presented in this paper (see below for details).

Creating Stimuli with FontWINDOW

FontWINDOW allows the user to create new fonts and edit existing fonts. Experimenters can use this software to create symbols and foreign language fonts, such as Arabic, Hebrew, Cyrillic, or Japanese Kanji, which are often not available with most microcomputer experimental software packages. Users can also create specialized fonts if they find existing fonts to be

inadequate for their needs. FontWINDOW also comes equipped with over 100 public domain fonts for immediate use, which include Arabic, Cyrillic, German Gothic script, Greek, Hebrew, and several Roman fonts containing characters with diacritical marks such as umlauts, chevrons, tildes, and accent marks. There are also several fonts containing icons such as arrows, trains, birds, flowers, stars, and moons, among other things that are included with the software. Users of FontWINDOW can also edit the fonts included with the software and those from other packages by importing them into FontWINDOW.

Whether one wishes to create a font or edit an existing one, the interface in FontWINDOW is easy to use. Characters can be selected from the Select Window, which displays the name of the font and the ASCII and hexadecimal values for each character. The selected character is displayed in the View Window at its normal resolution. Characters can be created and edited individually in the Edit Window similarly to many computerized drawing programs by using the functions displayed in the Tool Box. With the Tool Box, users can create lines, circles, and squares or draw freehand with the Pencil Tool. A Reference Line Tool can also be used as a guide for making character heights and widths consistent with other characters in the font. The selected character can also be moved within its spacing via the Move Tools. To change the size of the font, the Zoom function can be used to increase or decrease the heights and widths of all characters in the font. Unfortunately, smoothing of characters, such as O's and C's, is not automatic. This must be done within the Edit Window. If the user wishes to rotate all characters in the font, the Rotate tool can be used. The Tilt and Bold commands, when enabled, allow the user to create italicized and bolded fonts respectively. The default spacing for fonts is fixed spacing (i.e., thin characters such as "I" occupy the same amount of space as wide characters such as "W"). To change the font to proportional spacing (where characters are allotted space

according to their width), each character's width spacing must be adjusted by using the Width Control bar. Fortunately, MEL2 enables proportional spacing in fonts stored as having fixed spacing in FontWINDOW without having to make adjustments to each character individually (see below). Fonts created and edited in FontWINDOW are stored as *.fnt files, making them compatible with MEL2 and many other software packages.

Importing FontWINDOW Fonts Into MEL2

Several studies have used MEL2 and its predecessor, MEL (Schneider, 1990), to present stimuli in computer-controlled experiments (e.g., Besner & Stolz, 1999; Loring-Meier & Halpern, 1999; Stolz & Besner, 1997). In both versions of MEL, experiments can be programmed in either of two ways. Researchers using MEL2 can program the experiment using the MEL language code, just as if they were programming an experiment using the C or Pascal programming languages. The second way to create an experiment in MEL2 is to use the "form system." The form system is a more intuitive approach for inexperienced programmers. Each form contains blanks corresponding to the parameters of the experiment. The user simply needs to fill in the details (e.g., independent and dependent variables, presentation rate and order, instructions for the participant, etc.). Once the user has completed filling out the forms, MEL2 will generate the code for the experiment. For some experiments, the forms provided in this system are insufficient for the experimenter's needs. In this case, a separate code form can be created, with which the user writes the specific MEL programming code necessary for that part of the experiment.

In addition, stimuli can be presented in text or graphics mode. In text mode, stimuli are written to the screen more quickly than stimuli in graphics mode, but only standard text can be displayed. Graphics mode provides more display flexibility in that many different types of

displays can be used, including pictures and a wide variety of fonts. To enable graphics mode in MEL2, users can either use the graphics on/off option on a frame form or they can add a code form with the GRAPHICS_ON('') command. This command needs to be written only once as long as the experiment does not switch between graphics and text modes. For aesthetic reasons, the experimenter may wish to present word and sentence stimuli with proportional spacing. To do this, experimenters must use a code form if they wish to utilize the form system to create the experiment in MEL2. In the code form, the GRAPHICS_STYLE(128) command changes the font from fixed to proportional spacing. To load the font to be applied to the stimuli, the GRAPHICS_FONT command must be used. The stimuli to be displayed can then be written in the gray text window on a separate Frame Specifications form or it can be presented by using the DISPLAY command in the code form (see Appendix for use of these commands). Characters not found on the keyboard are entered by holding down the Alt key while entering the character's ASCII code. The experimenter can specify the position of stimuli by moving it within the gray text window of the Frame Specifications form, or the screen coordinates can be entered within the GRAPHICS_DISPLAY command for more precision.

An implementation of the techniques described in this paper is presented below to demonstrate an application of editing fonts in FontWINDOW for use in a MEL2 program. The study used the masked priming paradigm, which was chosen for the implementation of the described technique for two reasons. First, the masked repetition priming effect is a robust effect in that it has occurred in many published studies using a variety of prime durations and forward masks (e.g., Bodner & Masson, 1997; Forster & Davis, 1984; Gollan et al., 1997; Masson & Isaak, 1999). Thus, if the technique described here is to be used by other researchers, then it should yield roughly the same results (i.e., strong priming for words) as those found with other

methods used in repetition priming experiments. The second reason the masked priming paradigm was chosen is because we wanted to determine whether this technique could be used for presenting graphics fonts at an extremely short duration (i.e., 50 ms).

Two different stock fonts from FontWINDOW were chosen for prime and target stimuli, and a third font from the MEL2 directory (SYSTEM16.FNT) was used to present instructions and feedback. The forward mask and primes were presented using an edited version of "AFONT.FNT," whereas targets were presented with "BRENR29.FNT." AFONT.FNT was edited, in that the hash mark character was enlarged to more effectively mask the prime and to more closely resemble the target font such that it could be described as a fixation stimulus to participants. The new fonts were copied to the MEL2 directory and incorporated into the masked priming program. A condensed version of the MEL2 code is presented in the Appendix. The code presented indicates how to (1) load and display fonts with proportional spacing as it appears in the code form and (2) specify priming experiment parameters. In each section of the Appendix, comments—denoted by "!"—describe the functions of the code.

Implementation Demonstration: Repetition Priming in English

In the present masked repetition priming experiment, native English speakers were presented with letter strings for which they were instructed to make a lexical decision. The targets were primed with English words, which were either identical or unrelated to the target and presented for 50 ms following a forward mask presented for 500 ms.[1]

Method

Participants

Thirty-eight undergraduate psychology students at the University of Oklahoma, who were native speakers of English, participated in the study for course credit.

Apparatus, Materials, and Design

Stimuli were presented with a JDR Microcomputer (IBM/PC AT compatible), equipped with an EGA graphics card, on a Socos MA-1450 13-inch monitor. Two lists, each containing 128 prime-target pairs, were used. All primes were English words. For each list, 64 of the targets were words and 64 of the targets were non-words, created by changing one or two letters from American English words, matched in length with experimental stimuli, not used in the experiment. Both lists contained the same targets, which were presented once each. However, for one list, if a given target was preceded by an identical prime (e.g., Guitar-Guitar), it was preceded by an unrelated prime in the other list (e.g., Mouse-Guitar). Unrelated primes were taken from frequency-matched targets in the other list according to counts provided by Francis and Kučera (1982). This left both lists with the same overall frequency; frequency counts for targets preceded by identical primes were approximately equal in both lists (List 1 = 86.7, List 2 = 79.8). For non-word targets, related and unrelated conditions were created with respect to word length. That is, in the related prime-target condition, the prime had the same number of letters as the non-word target, whereas in the unrelated condition, the prime had a different number of letters. Primes were presented with a stock font from FontWINDOW (AFONT.FNT), in which the hash mark (i.e., #) character was enlarged to mask the prime more effectively. This mask was composed of a row of 13 hash marks. Targets were presented with an unedited stock font (BRENR29.FNT). For both primes and targets, the first letter was capitalized, whereas the rest of the word was written in lower case. Finally, responses were made using a five-key Psychology Software Tools Serial Response Box, which was set to a debounce rate of 1 ms.

Procedure

Stimuli were presented in the center of the computer screen. All prime-target pairs were immediately preceded by the forward mask, presented for 500 ms. Participants were told that this forward mask represented a cue signaling where and when the target was to appear. For each trial, the prime was presented for 50 ms immediately following the forward mask. The target was then presented until the participant responded. Participants were given feedback regarding their accuracy. Incorrect responses were followed by a "Wrong Response" message accompanied by a tone. Following trials ending in a correct response, participants were informed about their response time for that trial and their percent correct over all previous trials. Trials were presented in different random orders to each participant.

Participants were individually tested in a dimly lit cubicle. They were instructed to use the button box for their responses. "Yes" responses were made by pressing the middle response button with the index finger of their right hand and "no" responses were made by pressing the far right button with the middle finger of the same hand. Participants controlled the rate of trial presentation by pressing the far left button with their left index finger to begin the next trial. The instructions encouraged participants to respond as quickly as possible without compromising accuracy. Twelve practice trials were given to acquaint participants with the apparatus and the events of the experiment. At the conclusion of the experiment, participants were asked whether they were aware of the primes.

Results

Response time and accuracy data were analyzed for participants who reported awareness of no more than three primes. This led to the removal of seven participants from the data analysis. For participants who indicated that they could identify the prime only two or three

times (n = 3), response time (RT) and accuracy data corresponding to those trials were removed. Means and standard deviations for each participant and for each item were calculated for correct responses in each of the experimental conditions. To control for the influence of outliers, RTs that were more than two standard deviations greater or less than the mean for each participant were replaced with the corresponding cutoff RT. For participants (F_1) and items (F_2), separate analyses of variance (ANOVAs) were conducted. Three factors were included in the analyses: Target Lexicality (word vs. non-word), Prime-Target Relation (related vs. unrelated), and List Version (List 1 vs. List 2). Target Lexicality and Prime-Target Relation were repeated factors whereas List Version was a between-subjects factor. An alpha level of .05 was adopted as the significance level for both item and participant analyses.

Table 1 displays the results of the experiment. Lexical status determined whether priming occurred. Significant priming effects occurred for words, but not for non-words, resulting in a significant interaction between Lexicality and Prime-Target Relation in both analyses ($F_1(1, 30) = 46.13$, $p < .0001$; $F_2(1,62) = 8.69$, $p < 0.0045$). Response times to words preceded by identical primes were faster than those for words that were preceded by unrelated primes. No such facilitation was obtained for non-words. The main effects of Lexicality ($F_1(1, 30) = 90.42$, $p < .0001$; $F_2(1,62) = 204.15$, $p < 0.0001$) and Prime-Target relation ($F_1(1, 30) = 39.58$, $p < .0001$; $F_2(1,62) = 13.06$, $p < 0.001$) were also significant factors in both analyses. List was significant only in the items analysis ($F_1(1, 30) < 1$; $F_2(1,62) = 4.03$, $p = 0.049$) as participants receiving List 1 responded more quickly by approximately 20 ms (List 1 = 670.06, List 2 = 690.96).

General Discussion

The purpose of this paper was to present a method of presenting specialized graphics fonts in MEL2, a program designed for creating computer-controlled experiments. A masked

repetition priming experiment using two fonts from FontWINDOW was presented to validate this technique. Masked repetition priming effects comparable to those found in other masked priming experiments (cf. Forster & Davis, 1984; Gollan et al., 1997). In addition, these fonts could be presented at fast rates (here, 50 ms). Approximately three quarters of the total participants were unable to detect the primes at this presentation rate, a percentage comparable to that found by De Groot & Nas (1991). Thus, it seems that using graphics fonts edited with FontWINDOW to be used with MEL2 is a valid procedure, which yields results comparable to other studies.

Using FontWINDOW to develop graphics for MEL2 programs has several potential applications. Within the priming paradigm, experimenters can use FontWINDOW to create foreign-language fonts that they may not ordinarily find with normal programs for computer-controlled experiments. Such fonts could be applied to same-language priming experiments or to cross-language priming experiments with languages such as Arabic, Hebrew, Russian, or Chinese. In addition, if researchers are not satisfied with existing fonts that they have access to, FontWINDOW will allow them to either create their own font or to edit an existing font quite easily. For more general uses, experimenters can utilize FontWINDOW to create other forms of stimuli, such as pictures for children who do not read well or icons for picture naming experiments. Thus, using FontWINDOW for MEL2 experiments should benefit researchers looking for an easy way to present specialized stimuli in computer-controlled experiments.

References

Besner, D., & Stolz, J. A. (1999). Unconsciously controlled processing: The Stroop effect reconsidered. Psychonomic Bulletin & Review, 6, 449-455.

Bodner, G. E., & Masson, M. E. J. (1997). Masked repetition priming of words and nonwords: Evidence for a nonlexical basis for priming. Journal of Memory and Language, 37, 268-293.

Brown, H., Sharma, N. K., & Kirsner, K. (1984). The role of script and phonology in lexical representation. Quarterly Journal of Experimental Psychology, 36A, 491-505.

Cheung, H., & Chen, H.-C. (1998). Lexical and conceptual processing in Chinese-English bilinguals: Further evidence for asymmetry. Memory & Cognition, 26, 1002-1013.

Chen, H.-C., & Ng, M.-L. (1989). Semantic facilitation and translation priming effects in Chinese-English bilinguals. Memory & Cognition, 17, 454-462.

De Groot, A. M. B., & Nas, G. L. J. (1991). Lexical representation of cognates and noncognates in compound bilinguals. Journal of Memory and Language, 30, 90-123.

Evett, L. J., & Humphreys, G. W. (1981). The use of abstract graphemic information in lexical access. Quarterly Journal of Experimental Psychology, 33A, 325-350.

Forbach, G. B., Stanners, R. F., & Hochhaus, L. (1974). Repetition and practice effects in a lexical decision task. Memory & Cognition, 2, 337-339.

Forster, K. I., & Davis, C. (1984). Repetition priming and frequency attenuation in lexical access. Journal of Experimental Psychology: Learning, Memory, & Cognition, 10, 680-698.

Francis, W. N., & Kučera, H. (1982). Frequency analysis of English usage: Lexicon and grammar. Boston: Houghton Mifflin.

Gollan, T. H., Forster, K. I., & Frost, R. (1997). Translation priming with different scripts: Masked priming with cognates and non-cognates in Hebrew-English bilinguals. Journal of Experimental Psychology: Learning, Memory, and Cognition, 23, 1122-1139.

Keatley, C. W., Spinks, J. A., & De Gelder, B. (1994). Asymmetrical cross-language priming effects. Memory & Cognition, 22, 70-84.

Kirsner, K., & Smith, M. C. (1974). Modality effects in word identification. Memory & Cognition, 2, 637-640.

Kirsner, K., Smith, M. C., Lockhart, R. S., King, M. L., & Jain, M. (1984). The bilingual lexicon: Language-specific units in an integrated network. Journal of Verbal Learning and Verbal Behavior, 23, 519-539.

Loring-Meier, S., & Halpern, D. F. (1999). Sex differences in visuospatial working memory: Components of cognitive processing. Psychonomic Bulletin & Review, 6, 464-471.

Masson, M. E. J., & Isaak, M. I. (1999). Masked priming of words and nonwords in a naming task: Further evidence for a nonlexical basis for priming. Memory & Cognition, 27, 399-412.

Metagraphics Software. (1989). FontWINDOW: Graphics font editing toolkit (Version 4.1). Carmel, CA: Metagraphics Software Corporation.

Rayner, K., & Pollatsek, A. (1989). The psychology of reading. Englewood Cliffs, N. J.: Prentice-Hall.

Scarborough, D. L., Cortese, C., & Scarborough, H. S. (1977). Frequency and repetition effects in lexical memory. Journal of Experimental Psychology: Human Perception and Performance, 3, 1-17.

Schneider, W. (1990). MEL users' guide: Computer techniques for real time experimentation. Pittsburgh: Psychology Software Tools.

Schneider, W. (1995). MEL professional user's guide (Version 2.0). Pittsburgh: Psychology Software Tools.

Stolz, J. A., & Besner, D. (1997). Visual word recognition: Effort after meaning but not (necessarily) meaning after effort. Journal of Experimental Psychology: Human Perception and Performance, 23, 1314-1322.

Appendix

Condensed Version of MEL2 Code Used to Present Graphics Fonts in a Masked Repetition Priming Experiment

(1) Load and display fonts with proportional spacing as the code appears in the code form.

```
BEGIN_CODE_FORM
:255-60-1/ !================ CODE SUBROUTINE ================
! set & store block font file
DISPLAY_OFF                 !turn off display until characters are written
CLEAR                       !clear the screen
GRAPHICS_STYLE(128)         !enables proportional spacing
GRAPHICS_FONT('afont.fnt')  !font used for forward mask
GRAPHICS_DISPLAY( 190, 190, ' ############## ') !specifies where text should begin
:255-60-2/ !================ CODE SUBROUTINE ================
! set & store block font file
DISPLAY_OFF             !turn off display until characters are written
CLEAR                   !clear the screen
GRAPHICS_DISPLAY( 245, 190, FORM_TIE[3] ) !gets prime from insert file and ties
                                          with frame for specifications
:255-60-3/ !================ CODE SUBROUTINE ================
DISPLAY_OFF
CLEAR
GRAPHICS_FONT('brenr29.fnt') !font used for target stimuli
GRAPHICS_DISPLAY( 245, 190, FORM_TIE[4] ) !gets target from insert file and ties
```

with frame for specifications

```
:255-60-4/ !================ CODE SUBROUTINE ================
! return to small fonts to display feedback in standard characters
GRAPHICS_FONT('system16.fnt')
END_CODE_FORM    ! End of user code
```

(2) specify priming experiment parameters

```
! presents a series of trials of the task
:255-20-1/ ! Trial Specification #1
ZERO_SUM( FORM_TRIAL_MEAN_RT )
ZERO_SUM( FORM_TRIAL_MEAN_AC )
! sets Insert category }
FORM_INSERT_T = CHECK_INSERT ( FORM_BIE[2], 1, 255 )
FORM_TRIAL_TOTAL = TOTAL_STIMULUS( FORM_INSERT_T )
FORM_TRIAL_NUMBER = 1
WHILE (FORM_TRIAL_NUMBER <= FORM_TRIAL_TOTAL) DO BEGIN
 respAC = 0    !initializes dependent variables before each trial is recorded
 respSE = 0
 respRT = 0
 studyAC = 0
 studySE = 0
 studyRT = 0
DIVIDE_INSERTS( FORM_TIE, RANDOM_STIMULUS( FORM_INSERT_T,
TRIAL_INSERTS) )
```

```
!sets independent variable values
        wordnon = CHECK_INSERT ( FORM_tIE[1], -32766, 32767 )    !Target lexicality
        relnon = CHECK_INSERT ( FORM_tIE[2], -32766, 32767 )     !Prime-target relation
        tag = CHECK_INSERT ( FORM_tIE[5], -32766, 32767 )  !References prime-target pair
        FORM_MISC_INSERT = ""  PERFORM( "255-50-20" )  ! Get ready display
        FORM_MISC_INSERT = ""  PERFORM( "255-60-1" )  ! graphic on/give instruct
        FORM_MISC_INSERT = ""  PERFORM( "255-50-30" )  ! mask specs
        FORM_MISC_INSERT = ""  PERFORM( "255-60-2" )  ! center prime
        FORM_MISC_INSERT = ""  PERFORM( "255-50-40" )  ! prime specs
        FORM_MISC_INSERT = ""  PERFORM( "255-60-3" )  ! center target
        FORM_MISC_INSERT = ""  PERFORM( "255-50-50" )  ! collect response to target
        FORM_MISC_INSERT = ""  PERFORM( "255-60-4" )  ! return to small font
!runs through trials until inserts are exhausted
        EXECUTE
        if LENGTH( FORM_FEEDBACK_LIST ) > 0 then
                PERFORM( "255-99-1" )  !perform feedback if needed
        LOG_TRIAL
        FORM_TRIAL_NUMBER = FORM_TRIAL_NUMBER + 1
        END                                    !end of trials
!presents the forward masking stimulus in afont.fnt
        :255-50-30/ ! Frame Specification #30
        DISPLAY_OFF
        WAIT_TOP
```

```
        DISPLAY_ON
        LATENCY( studyRT )
        SRBOX_IN( FORM_KEY[30], "1", "M" )
        WAIT( 500 )                    !displays forward mask for 500 ms
!presents prime for 50 ms in afont.fnt
        :255-50-40/ ! Frame Specification #40
        DISPLAY_OFF
        WAIT_TOP
        DISPLAY_ON
        WAIT( 50 )              ! presents prime for 50 ms
!present target for 15000 ms or response in brenr29.fnt
        :255-50-50/ ! Frame Specification #50
        DISPLAY_OFF
        WAIT_TOP
        DISPLAY_ON
        LATENCY( respRT )
        SRBOX_IN( FORM_KEY[50], "35", "MW" )
        WAIT( 15000 )
```

Authors' Note

This research was conducted in partial fulfillment of a M.S. degree at the University of Oklahoma by the first author. Monetary support for FontWINDOW software was provided by a grant from the Graduate Student Senate at the University of Oklahoma. Correspondence can be addressed to Brian Friel, Department of Psychology, University of Oklahoma, 455 West Lindsey Street, Norman, Oklahoma 73019. Electronic mail correspondence can be sent to bfriel@ou.edu.

Footnote

[1]The implementation presented is similar to Gollan et al.'s (1997) English within-language condition.

Table 1

Mean Lexical Decision Times (in ms) as a Function of Target Lexicality and Prime-Target Relation

	Target Lexicality	
Prime-Target Relation	Words	Non-Words
Related	584	746
Unrelated	642	750
Priming Effect	58	4